Alice Lai
15-462 Computational Photography

1. **Bilateral Filtering**
(**Note**: I did all of my parameter tuning on the ambient images)

Basic Bilateral Filtering

	$\sigma_r = 0.05$	$\sigma_r = 0.1$	$\sigma_r = 0.45$
$\sigma_s = 25$			
$\sigma_s = 45$			

- Making σ_r large made the images much blurrier (which makes sense given that σ_r is the parameter for the spatial kernel). I decided that I liked the image best when $\sigma_r = 0.1$.
- I didn't notice too much of a difference in the image when i manipulated σ_s so I decided that my σ_s parameter would just be the one provided (25).
- **Advantages:**
    - The dark patches on the wall are completely eliminated by the Gaussian filter
    - There are not that many light reflections on the wall anymore
- **Disadvantages:**
    - The edges of the speaker-like objects in the back right got blurred out by the Gaussian filter; the shadows look unnaturally blurred.
    - Some slight visual artifacts at the edge between illuminated and dark regions
- Final parameters: $\sigma_s = 25$, $\sigma_r = 0.1$

Joint Bilateral Filtering

	$\sigma_r = 0.05$	$\sigma_r = 0.1$
$\sigma_s = 3$		
$\sigma_s = 7$		
$\sigma_s = 11$		

- Parameters for joint bilateral filtering were quite similar to vanilla bilateral filtering, save for the spatial kernel. The drop in range of possible values make sense since [blah].
- **Advantages:**
  - Where the vanilla bilateral filter failed to preserve the detail of the edges around the speaker-like objects in the background, the joint bilateral filter recovered that information.
- **Disadvantages:**
  - The light reflections on the wall are quite pronounced.
  - The dark patches on the wall are visible.
- Final parameters: $\sigma_s = 3$, $\sigma_r = 0.1$

Denoising with Detail Transfer

$\varepsilon = 0.02$	$\varepsilon = 0.5$	$\varepsilon = 2.0$	$\varepsilon = 10.0$
			

- To determine the optimal parameters for Detail Transfer algorithm, I used the flash bilateral filter and ambient joint filter images with the optimal parameters that I determined for those algorithms
- I saw no differences between ε = 0.5 and ε = 10.0 (with no visual artifacts), so I went with ε = 1.0 to make the algorithm more computationally cheap.
- I was unable to discern any visible difference between the detail transfer image and the joint bilateral images.
- Final parameter: ε = 1.0

Mask-Based Merging

ε= 0.02	ε = 0.05	ε = 0.2	ε = 0.5
			

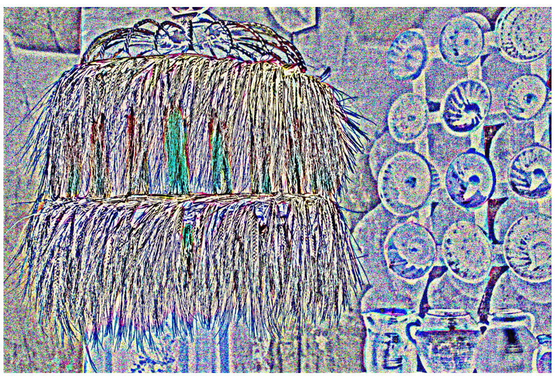
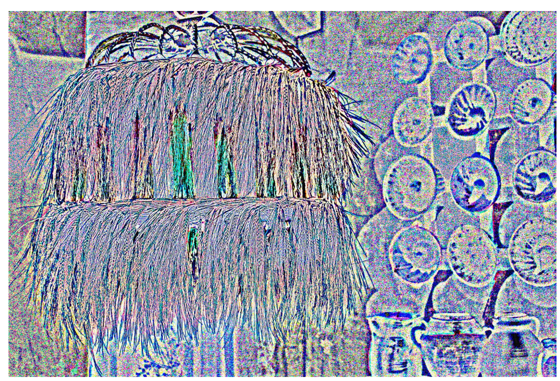
- I liked ε = 0.05 the most, since that image had the optimal balance of removal of dark wall patches and preservation of detail around the speakers in the background
- **Advantages:**
  - Combines the advantages of bilateral filter and joint bilateral filter/detail transfer in regards to preserving sufficient detail while smoothing out undesirable scene artifacts
- **Disadvantages:**
  - Still does not quite preserve the detail around the speakers
- Final parameters: ε = 0.05

Final Four Images

Vanilla Bilateral Filter	Joint Bilateral Filter
	
Detail Transfer	Mask-Based Merging
	

Detailed Comparison

I took the difference between the original ambient image and the final images of the four algorithms: bilateral filter, joint bilateral filter, detail transfer, and mask merge. The images for Detail Transfer and Joint Bilateral Filter were very similar to each other, those two images preserved a lot of detail around the fringe/fur lamp and the speakers from the original image. The images for Mask Merge and Vanilla Bilateral Filter were similar to each other as well. Where these two images lacked in detail in the fringe/fur lamp and the speakers, they had no dark patches on the walls or any undesirable artifact from the original scene. I thought the **Mask Merge** image was the best out of all 4 of the output images, with the most optimal balance of visual artifact smoothing and detail preservation.

Vanilla Bilateral Filter (Diff)	Joint Bilateral Filter (Diff)
	
Detail Transfer (Diff)	Mask Merge (Diff)
	

2. Gradient-domain processing

	$\nabla\alpha$ gradient field	$\nabla\Phi$ gradient field	$\nabla\Phi^{*}$ gradient field $\sigma = 40$, $\tau_s = 0.9$
x			
y			

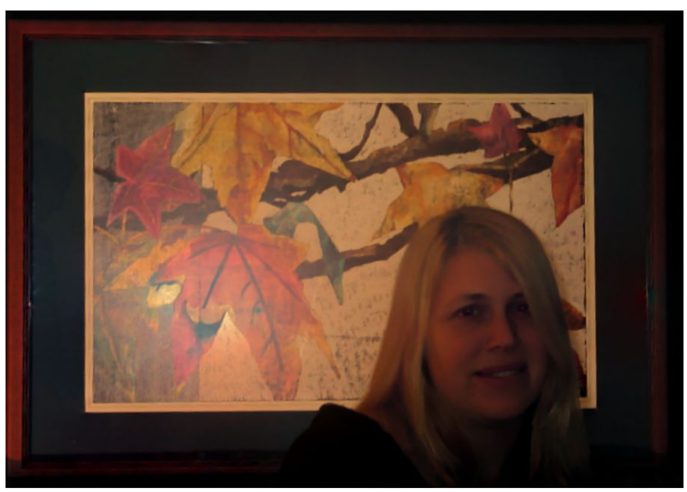
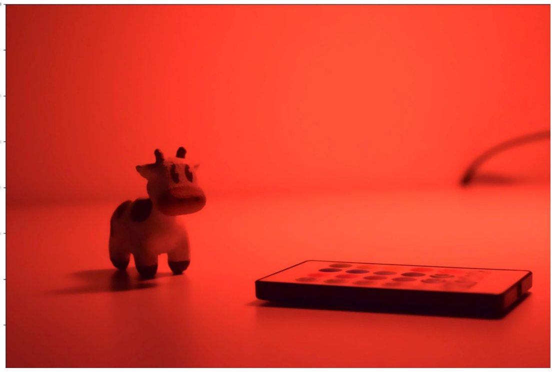


Figure 1: Final Fused Image

3. **Capture your own flash/no-flash pairs**

Bilateral Filtering Denoising	Gradient Domain-Processing Fusion
	

Note: I used JPG files for both of my flash/no-flash pairs. For the gradient domain-processing function, I downsampled the flash/no-flash pair by a factor of 9 (3 on each dimension) for the gradient fusion algorithm in order for my script to consume a reasonable amount of memory of my machine.